

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Serial No.: Not Yet Assigned
Filed: April 15, 1998
Entitled: METHOD AND APPARATUS FOR REAL TIME NETWORK
COMMUNICATION
Group No.: Not Yet Assigned
Examiner: Not Yet Assigned
Inventors: Andrew T. Busey and Gerald H. Weghorst, Jr.

Assistant Commissioner for Patents
Washington, D.C. 20231

April 15, 1998

TRANSMITTAL LETTER

Dear Sir:

Applicant encloses herewith for filing:

- ☒ Continuation Patent Application consisting of a Specification on 30 sheets and an Abstract;
- ☒ Three (3) Pages of Drawings for Figures 1-3;
- ☒ Certificate of Mailing by Express Mail (37 CFR 1.10); and
- ☒ A self-addressed, postage pre-paid postcard to be returned to Applicant.

Missing Parts to this application will be filed in accordance with the Notice of Missing Parts upon receipt of same by Applicant.

Respectfully submitted,

Brobeck, Phleger & Harrison LLP



Benjamin C. Huang., Reg. No. 41,080

Brobeck, Phleger & Harrison LLP
301 Congress Avenue, Suite 1200
Austin, Texas 78701
(512) 479-2948

I certify that this correspondence is being deposited with the United States Postal Service as Express Mail in an envelope addressed to: Assistant Commissioner for Patents, Washington, DC 20231 on April 15, 1998.

Signature of person making deposit


Cynthia Luna

I hereby certify that this correspondence is being deposited with the United States Postal Service as express mail in an envelope addressed to: BOX PATENT APPLICATION, Assistant Commissioner for Patents, Washington, D.C. 20231, on

April 15, 1998, Express Mail No: _____

EM037897855US

4-15-98

Date of Signature

Carthine Luna
Signature of Person Making the Deposit

1 **METHOD AND APPARATUS FOR REAL TIME NETWORK COMMUNICATION**

2 **Andrew T. Busey**
3 **Gerald H. Weghorst, Jr.**

4
5 **CROSS-REFERENCE TO RELATED APPLICATION**

6 This patent application is a continuation of coassigned copending United States
7 Patent Application Serial No. 08/722,898, filed September 27, 1996, which is hereby fully
8 incorporated herein by this reference thereto.

9 **BACKGROUND**

10 **Technical Field**

11 This patent application relates in general to network communication, and in
12 particular to real-time network communication.

Description of Related Art

The location and exchange of data over computer networks is controlled by various network protocol. For example, the World Wide Web (hereinafter "Web") is a system of communications protocols that presents information in documents that are capable of being linked to other documents. The documents are stored in a distributed manner across the Internet on the networked computers, and are accessed using programs known as browsers.

The Web is a system of protocols exchanged between a host computer running an application, known as a server, that delivers Web documents, and a user's computer, known as the client. The most fundamental of Internet protocols is Transmission Control Protocol/Internet Protocol, or TCP/IP, which in effect provides a description of the document data being exchanged and a destination for that data.

Web documents are created using a markup language known as html, or Hypertext Markup Language. Generally, a markup language is a set of instructions, or markups that is used to direct a browser how to display and manage a document. Specifically, html defines the format of a Web document and enables hyperlinks to be embedded in the Web document. Hyperlinks are used to connect a document on one host computer to a document on another host computer. The following html paragraph is illustrative.

<P>

Welcome to the home page of ichat, Inc.. We develop software that expands the
functionality and accessibility of real-time Internet chat systems.

The html tags " ... " instructs the browser to display ichat Inc. in bold font. The html tag "<A HREF=" instructs the browser to create a link to a web page referenced by the embedded Uniform Resource Locator ("URL"), which is a type of address, and to use the

25 word "software" embedded between the tags "> ... " as the hyperlinked word. The link
26 may be a target, which is a word or phrase in another section of the same Web page; a
27 relative link, which is another Web page within the current site, either forward or backward
28 relative to the current page; or an external or absolute link, which is a Web page on another
29 host.

30 A Web browser or client requests a Web document in accordance with its URL,
31 which typically is furnished either manually by the user, through a Web browser bookmark,
32 or through a hyperlink embedded in an html document. The Web browser causes a two-way
33 TCP/IP connection to be established between the client and the host from which the desired
34 Web document is available, and then generates and sends to the host a request header to
35 establish an HTTP connection. The server on the host responds to the URL either by (i)
36 following the directory path contained in the URL and opening the file containing the
37 requested document; or (ii) running a CGI, or Common Gateway Interface, script; or (iii)
38 detecting an error and generating an error document. The host then returns the document
39 along with its file type to the client. After the client acknowledges receipt of the Web
40 document, the host closes the HTTP connection even though the TCP/IP connection may be
41 maintained by the service provider.

42 The term "HTTP" used above stands for Hypertext Transfer Protocol, which is a
43 transfer protocol that sits on top of TCP/IP and is a stateless protocol designed to transfer
44 documents at a high rate of speed. As a stateless system, HTTP does not retain any
45 information from one document transfer to the next. If additional documents are needed,
46 each additional document must be transferred by opening a new HTTP connection,
47 requesting the document, delivering the document, and closing the connection.

48 After the requested Web document is received and the HTTP connection closed, the
49 browser interprets the page format from the imbedded html tags in a process known as
50 parsing. Typically, the html tags format text, load images, and embeds hyperlinks. When the

51 browser encounters an html tag that links to a different type of MIME file such as a sound
52 clip or picture, the browser separately requests that sound clip or picture in another HTTP
53 connection. Hence, if the Web page contains both a sound clip and an image, three requests
54 are made - one for the original html page, one for the sound clip file, and one for the image
55 file.

56 Although generally successful for many Web functions, the HTTP protocol is
57 ineffective for enabling real-time functions such as chat over networks such as the Internet.
58 However, chat is available over the Internet using the Internet Relay Chat protocol, or IRC.
59 IRC uses the TCP/IP protocol in a client/server model. IRC client software is installed on the
60 user side and integrated into the Web browser. One client initiates a channel by connecting
61 to an IRC enabled server, which may or may not be on the same host as the Web server.
62 Other clients join the channel by typing a join IRC command. The IRC server mediates the
63 channel, passing each message to all channel members or to particular channel members, as
64 determined by the originator of the message.

65 While IRC has had some success, its widespread use is hampered by several
66 inconvenient aspects of the protocol. For example, IRC is cumbersome and does not support
67 features common in Web browsers, such as hyperlinks and graphics.

SUMMARY

A real time communications protocol connection is formed over a network communications connection. A markup language instruction is embedded in a message. The message is sent on the real time communications protocol connection.

It is a technical advantage that real-time communication is maintainable over computer networks such as the Internet with enhanced functionality of the type generally available to World Wide Web users using Web browsers.

1

BRIEF DESCRIPTION OF THE DRAWINGS

2

In the drawings, in which like reference characters indicate like parts:

3

4

Figure 1 is a schematic diagram of network protocol connections between clients and a host;

5

Figure 2 is a flow chart of a method for real time network chat; and

6

7

Figure 3 is a schematic diagram of how a hyperlink functions during a real time network chat.

DETAILED DESCRIPTION

Figure 1 and Figure 2 show a process for real-time conferencing across the Internet. The process begins with a user who launches a chat session from his or her computer, preferably from a browser application running on the computer, by running an application called a real time markup ("RTM") chat client. The computer operating system ("OS") causes a two-way TCP/IP connection to be established between the client computer and a host computer for the chat session, while the RTM chat client causes a real time full duplex connection to be established between the RTM chat client and a real time server on the host. Other users join the chat session by establishing TCP/IP connections and launching their own RTM chat clients. Figure 1 shows three RTM chat clients 110, 120 and 130, which run on top of respective TCP/IP clients 112, 122 and 132. The TCP/IP connections are established with a host computer, which runs TCP/IP host software 140 and typically hosts several different types of servers. Figure 1 illustratively shows four servers, an HTTP server 142, a telnet/chat server 144, an FTP server 146, and an Internet Relay Chat ("IRC") server 148. Typically, a variety of other server types reside on the host computer as well, including, for example, Gopher, Usenet and WAIS.

A real time chat client is any client capable of sustaining what appears to a user to be real time chat. The effect of real time is created by using a continuously open connection protocol such as, for example, a continuously open streaming protocol such as telnet or a continuously open connection packet protocol such as IRC. Telnet is a well known streaming protocol used to establish bi-directional continuously opened sockets and full duplex data transmission to achieve real time communications. The telnet protocol is an industry standard. UNIX hosts are generally provided with telnet servers as part of their operating systems. Other examples of continuously opened connection streaming protocols include UDP, or Universal Data Protocol, and a variety of proprietary protocols. IRC is a well-known packet protocol used to establish bi-directional continuously opened sockets and

27 full duplex data transmission to achieve real time communications. The IRC protocol is an
28 industry standard, fully defined in RFC 1459. In contrast, the HTTP protocol defines a
29 transactional half-duplex data transmission. HTTP connections are opened and closed as
30 documents are requested and sent. Real time communication is not realized.

31 A markup language is any language that enables document formats to be defined,
32 and may also enable hyperlinks to be embedded in documents. The most popular markup
33 language in use on the Web is html, which supports embedded hyperlinks, various font
34 styles such as bold and italics, and various MIME (Multipurpose Internet Mail Extension)
35 file types for text and embedded graphics, video and audio.

36 Figure 2 shows what happens when a RTM chat client is launched. Illustratively, the
37 chat client in Figure 2 is a telnet html chat client and the host includes a telnet server and a
38 server-side application known as a chat server that enables communication between two or
39 more chat clients. While Figure 2 shows use of the telnet protocol and a compatible chat
40 server, the IRC protocol and an IRC chat server may be used if desired, as well as any other
41 continuously open bi-directional connection chat client - server types and compatible chat
42 server applications. Chat servers are well known; for example, the telnet protocol and
43 proprietary chat server software is commonly used by commercial BBS services, and the
44 IRC protocol and IRC server side chat applications are common in many UNIX
45 environments. While Figure 2 also shows use of html, other markup languages may be used
46 if desired.

47 After the TCP/IP and telnet connections are made (step 200), the telnet html chat
48 client immediately begins to receive any messages being posted by the chat server, and may
49 send messages to other telnet html chat clients through the chat server or remain idle in the
50 event that no messages are being sent or received. While non-html telnet clients may also be
51 connected to the chat server, they will not be capable of displaying the incoming data with
52 fidelity because they will not be able to properly parse it.

53 Messages outgoing from the telnet chat client are processed as follows. The telnet
54 chat client is designed either to send each keystroke to the host either individually or in
55 groups. In either case, the telnet chat client appends the keystroke(s) to a TCP/IP header and
56 the resulting packet is sent to the chat host (step 220). The chat host parses the incoming
57 data in real time (step 222). If the chat host detects a telnet escape sequence (step 224), it
58 processes the detected escape sequence (step 226). Otherwise, the chat host simply posts the
59 data (step 230) to all connected telnet clients or to a specific or ones of connected telnet
60 clients if so instructed by the chat server. The telnet host does not recognize or process html
61 tags in the data, and simply posts them. Connected telnet clients that are not html enabled
62 simply display the html tags as they are received. However, connected telnet html clients
63 recognize and respond to the html tags in the data.

64 Messages outgoing from an IRC chat client are processed in a slightly different
65 manner. An IRC packet is the entire series of keystrokes preceding a carriage return. An IRC
66 chat client appends the IRC packet or in some cases breaks up the IRC packet into sub-
67 packets and appends each sub-packet to a TCP/IP header, and the resulting TCP/IP packet is
68 sent to the IRC chat host. The IRC chat host parses the incoming data in real time,
69 processing any IRC headers and handling the appended data accordingly.

70 The telnet chat client processes incoming messages containing html tags as follows.
71 The telnet chat client parses the incoming data (step 210) to distinguish between html tags
72 and characters to be displayed. If an html tag is detected (step 212), the tag is processed as
73 appropriate (step 214). If an html tag is not detected (step 212), the incoming data is
74 displayed on the chat screen of the telnet chat client computer (step 216). In either case, the
75 telnet chat client then looks for more data to process (step 218), and either resumes parsing
76 or idles if no incoming or outgoing message is present.

77 The telnet connection is terminated either by the client or the host. Termination is
78 done by releasing the socket for the connection, in a manner well known in the art.

79 An example of a real time chat session among chat clients using html is as follows.

80 <Sarah> Hi everyone! I found a *great* web site. **Check out** the ichat site.

81 <Sam> Thanks for the info, Sara. I'm going to check out the site now. Bye.

82 This text appears on the screens of the html chat clients who are members of the chat
83 session.

84 When Sarah types her message, she uses either macros or html itself to cause the
85 word "great" to appear in an italics font style, the phrase "Check out" to appear in a bold
86 font style, and to create the hyperlink ichat site. Sarah's chat client software sends the
87 following illustrative data stream to members of the chat session via the host.

88 Hi everyone! I found a <I> great</I> web site. Check out the <A
89 HREF="http://www.ichat.com"> ichat site.

90 The html chat client software displays Sara's message as it is typed in a normal font, until
91 the "<I>" tag is detected. The characters "great" are displayed as they are typed in an italics
92 style font until the "</I>" tag is detected, after which subsequent characters are again
93 displayed as they are typed in a normal font. When the "" tag is detected, the subsequent
94 characters "great" are displayed as they are typed in a bold font until the "" tag is
95 detected, after which subsequent characters are again displayed as they are typed in a normal
96 font. When the tag "" is detected, Sam's software
97 responds by linking the URL "http://www.ichat.com" to the text following the tag, until the
98 tag "" is detected. Hence, the URL "http://www.ichat.com" is linked to the hyperlink
99 ichat site. This hyperlink is displayed as its characters are typed in a underlined and colored
100 font until the "" tag is detected, after which any subsequent characters are displayed at
101 they are typed in a normal font.

102 Sam responds to Sara's message with his message, and then simply clicks on the
103 hyperlink "ichat site" in his chat window using either his mouse or keyboard navigation.
104 This action launches Sam's Web browser, if it is not already running. Sam's Web browser
105 takes him to the ichat home page, without need for Sam to enter a URL.

106 The manner in which hyperlinks function in a chat session among RTM chat clients
107 is shown in more detail in Figure 3. The two-way arrow between RTM chat client 314 in
108 client 310 and a real time server 324 in host 320 represents a bi-directional TCP/IP - real
109 time protocol communications channel. The two-way arrow between RTM chat client 334 in
110 client 330 and the real time server 324 in host 320 also represents another bi-directional
111 TCP/IP - real time protocol communications channel. The one way arrows between web
112 browser 332 in the client 330 and HTTP server 342 in host 340 represent respective one way
113 TCP/IP HTTP (transactional) protocol communications channels. The host 310 need not
114 include a Web browser, the host 320 need not include an HTTP server 322, and the host 340
115 need not include a real time server 340.

116 RTM chat client 314 (*e.g.* Sarah) creates a message that includes an embedded
117 hyperlink, and sends that message through the real time server 324 (action "A") to the RTM
118 chat client 334 (*e.g.* Sam) (action "B"). Note that other actions that may be occurring, such
119 as echo of the message back to the RTM chat client 314 and communication of the message
120 to other joined chat clients, are omitted for clarity. The client 330 (*e.g.* Sam) then causes his
121 Web browser 332 to access the URL associated with the hyperlink embedded in the chat
122 message (*e.g.* ichat site) (action "C"). Action "C" is performed in any suitable manner. For
123 example, if the Web browser 332 is inactive, the RTM chat client 334 simply launches the
124 Web browser 332 using the URL associated with the hyperlink as a command line argument.
125 If the Web browser 332 happens to be running, the RTM chat client 334 communicates the
126 page request to the Web browser 332 using any suitable interface protocol such as the DDE
127 protocol, which is standard in such operating systems as the Microsoft® Windows® Version
128 3.1 operating system and the Microsoft® Windows® 95 operating system. Newer protocols

129 and methods suitable for having the RTM chat client 334 cause the Web browser 332 to
130 acquire a Web page include plug-in technologies, ActiveX technologies, and Java
131 technologies. The Web browser 332 makes a TCP/IP connection with the HTTP server 342
132 (or any other HTTP server, including HTTP server 322) and Web browser 332 makes a
133 request for a Web page (action "D") by sending the URL associated with the embedded
134 hyperlink. The HTTP server 342 responds by delivering the requested Web page (action
135 "E"), and the TCP/IP connection between the Web Browser 332 and the HTTP server 342 is
136 terminated. Meanwhile, the bi-directional TCP/IP - real time protocol communications
137 channels between the RTM chat client 314 and the real time server 324, and between the
138 RTM chat client 334 and the real time server 324 remain open if desired to continue the chat
139 session.

140 Although an illustrative embodiment and its advantages have been described in detail
141 hereinabove, they have been described as example and not as limitation. Various changes,
142 substitutions and alterations can be made in the illustrative embodiment without departing
143 from the breadth, scope and spirit of the claims. For example, various real time
144 communications protocols and various markup languages may be used.

CLAIMS

What is claimed is:

- 1 1. A method for real time network communication, comprising:
2 forming a real time communications protocol connection over a network
3 communications connection;
4 embedding a markup language instruction in a message; and
5 sending the message on the real time communications protocol connection.
- 1 2. The method of Claim 1 wherein the embedding comprises embedding a markup
2 language instruction in the message, the markup language instruction being a hyperlink
3 instruction.
- 1 3. The method of Claim 1 wherein:
2 the forming comprises forming a real time continuously open bi-directional
3 communications protocol connection; and
4 the sending comprises sending the message on the real time continuously open bi-
5 directional communications protocol connection.
- 1 4. The method of Claim 1 wherein:
2 the forming comprises forming a real time chat communications protocol connection;
3 and
4 the sending comprises sending the message on the real time chat communications
5 protocol connection.

1 5. A method for real time network communication, comprising:
2 forming a real time communications protocol connection over a network
3 communications connection;
4 receiving a first message on the real time communications protocol connection, the
5 first message including a hyperlink instruction;
6 parsing the first message to identify the hyperlink instruction included therein; and
7 displaying the first message in accordance with the hyperlink instruction included
8 therein.

1 6. The method of Claim 5 and comprising:
2 embedding a hyperlink instruction in a second message; and
3 sending the second message on the real time communications protocol connection.

1 7. The method of Claim 5 wherein the receiving comprises receiving the first
2 message on the real time communication protocol connection, the hyperlink instruction
3 being associated with a document address,
4 and comprising:
5 passing the document address to a document acquisition apparatus.

1 8. The method of Claim 5 wherein:
2 the forming comprises forming a real time chat communications protocol connection;
3 and
4 the receiving comprises receiving the first message on the real time chat
5 communications protocol connection.

[illegible]

1 10. A method for real-time network communication, wherein the network includes
2 TCP/IP connections formed between a plurality of clients and a host, and respective real
3 time communications protocol connections formed over the TCP/IP connections, the method
4 comprising:

5 receiving a message including a hyperlink language instruction from the host through
6 at least one of the real time communications protocol connections;
7 parsing the message in the client sent the message by the host; and
8 displaying the message in the client sent the message by the host in accordance with
9 the hyperlink language instruction included therein.

1 11. The method of Claim 10 wherein the receiving comprises receiving the message
2 including the hyperlink language instruction, the hyperlink language instruction being
3 associated with a document address,

4 and comprising:
5 passing the document address to a document acquisition apparatus.

1 12. The method of Claim 10 wherein the real time communications protocol
2 connections are real time chat communications protocol connections, and wherein the
3 receiving comprises receiving the message from the host through at least one of the real time
4 chat communications protocol connections.

1 13. The method of Claim 10 wherein the real time communications protocol
2 connections are real time continuously open bi-directional communications protocol
3 connections, and wherein the receiving comprises receiving the message from the host
4 through at least one of the real time continuously open bi-directional communications
5 protocol connections.

1 14. A method for real time network communication, comprising:
2 forming a real time communications protocol connection over a network
3 communications connection;
4 embedding a markup language instruction in a message; and
5 sending the message on the real time communications protocol connection.

1 15. The method of Claim 14 wherein the embedding comprises embedding the
2 markup language instruction, the markup language instruction being an html instruction.

1 16. The method of Claim 15 wherein the embedding comprises embedding the html
2 instruction, the html instruction being a hyperlink instruction.

1 17. The method of Claim 16 wherein the embedding comprises embedding the
2 hyperlink instruction, the hyperlink instruction being associated with a URL,
3 and comprising:
4 passing the URL to a Web browser.

1 18. The method of Claim 15 wherein the embedding comprises embedding the html
2 instruction, the html instruction being a bold tag.

1 19. The method of Claim 15 wherein the embedding comprises embedding the html
2 instruction, the html instruction being an italics tag.

1 20. The method of Claim 14 wherein:
2 the forming comprises forming a real time chat communications protocol connection;
3 and
4 the sending comprises sending the message on the real time chat communications
5 protocol connection.

- 1 21. The method of Claim 14 wherein:
- 2 the forming comprises forming a real time continuously open bi-directional
- 3 communications protocol connection; and
- 4 the sending comprises sending the message on the real time continuously open bi-
- 5 directional communications protocol connection.

1 22. A method for real time network communication, comprising:
2 forming a real time communications protocol connection over a network
3 communications connection;
4 receiving a first message on the real time communications protocol connection, the
5 first message including a markup language instruction;
6 parsing the first message to identify the markup language instruction included
7 therein; and
8 displaying the first message in accordance with the markup language instruction
9 included therein.

1 23. The method of Claim 22 wherein the receiving comprises receiving the first
2 message, the markup language instruction being a hyperlink instruction.

1 24. The method of Claim 23 wherein the receiving comprises receiving the first
2 message, the hyperlink instruction being associated with a URL;
3 and comprising:
4 passing the URL to a Web browser.

1 25. The method of Claim 22 wherein the receiving comprises receiving the first
2 message, the markup language instruction being a bold tag.

1 26. The method of Claim 22 wherein the receiving comprises receiving the first
2 message, the markup language instruction being an italics tag.

1 27. The method of Claim 22 wherein:
2 the forming comprises forming a real time chat communications protocol connection;
3 and
4 the receiving comprises receiving the first message on the real time chat
5 communications protocol connection.

- 1 28. The method of Claim 22 wherein:
- 2 the forming comprises forming a real time continuously open bi-directional
- 3 communications protocol connection; and
- 4 the receiving comprises receiving the first message on the real time continuously
- 5 open bi-directional communications protocol connection.

1 29. A communication client, comprising:
2 a computer for:
3 forming a real time communications protocol connection over a network
4 communications connection;
5 embedding a markup language instruction in a message; and
6 sending the message on the real time communications protocol connection.

1 30. The communication client of Claim 29 wherein the markup language instruction
2 is a hyperlink instruction.

1 31. The communication client of Claim 29 wherein the real time communications
2 protocol connection is a real time chat communications protocol connection.

1 32. The communication client of Claim 29 wherein the real time communications
2 protocol connection is a real time continuously open bi-directional communications protocol
3 connection.

1 33. A communication client, comprising:
2 a computer for:
3 forming a real time communications protocol connection over a
4 communications connection;
5 receiving a message on the real time communications protocol connection;
6 parsing the message to identify a markup language instruction included
7 therein; and
8 displaying the message in accordance with the markup language instruction
9 included therein.

1 34. The communication client of Claim 33 wherein the markup language instruction
2 is a hyperlink instruction.

1 35. The communication client of Claim 33, wherein the message is a first message,
2 wherein the markup language instruction is a first markup language instruction, and wherein
3 the computer is for:
4 sending a second message on the real time communications protocol connection; and
5 embedding a second markup language instruction in the second message.

1 36. The communication client of Claim 33 wherein the markup language instruction
2 is associated with a document address, and wherein the computer is for passing the
3 document address to a document acquisition apparatus.

1 37. The communication client of Claim 33 wherein the real time communications
2 protocol connection is a real time chat communications protocol connection.

1 38. The communication client of Claim 33 wherein the real time communications
2 protocol connection is a real time continuously open bi-directional communications protocol
3 connection.

1 39. A communication server, comprising:

2 a computer for:

3 forming a real time communications protocol connection over a network
4 communications connection;

5 receiving a message on the real time communications protocol connection,

6 wherein the message includes a markup language instruction.

1 40. The communication server of Claim 39 wherein the markup language instruction

2 is a hyperlink instruction.

1 41. The communication server of Claim 39 wherein the computer is for:

2 receiving the message from a first communication client; and

3 sending the message on the real time communications protocol connection to a

4 second communication client.

1 42. The communication server of Claim 39, wherein the message is a first message,

2 wherein the markup language instruction is a first markup language instruction, and wherein

3 the computer is for receiving a second message on the real time communications protocol

4 connection, wherein the second message includes a second markup language instruction.

1 43. The communication server of Claim 39 wherein the markup language instruction

2 is associated with a document address, and wherein the computer is for passing the

3 document address to a document acquisition apparatus.

1 44. The communication server of Claim 39 wherein the real time communications

2 protocol connection is a real time chat communications protocol connection.

1 45. The communication server of Claim 39 wherein the real time communications

2 protocol connection is a real time continuously open bi-directional communications protocol

3 connection.

1 46. A computer program product, comprising:
2 a computer application processable by a computer for causing the computer to:
3 form a real time communications protocol connection over a network
4 communications connection;
5 embed a markup language instruction in a message; and
6 send the message on the real time communications protocol connection; and
7 apparatus from which the computer application is accessible by the computer.

1 47. The computer program product of Claim 46 wherein the markup language
2 instruction is a hyperlink instruction.

1 48. The computer program product of Claim 46 wherein the real time
2 communications protocol connection is a real time chat communications protocol connection

1 49. The computer program product of Claim 46 wherein the real time
2 communications protocol connection is a real time continuously open bi-directional
3 communications protocol connection.

1 50. A computer program product, comprising:
2 a computer application processable by a computer for causing the computer to:
3 form a real time communications protocol connection over a network
4 communications connection;
5 receive a first message on the real time communications protocol connection,
6 the first message including a markup language instruction;
7 parse the first message to identify the markup language instruction included
8 therein; and
9 display the first message in accordance with the markup language instruction
10 included therein; and
11 apparatus from which the computer application is accessible by the computer.

1 51. The computer program product of Claim 50 wherein the markup language
2 instruction is a hyperlink instruction.

1 52. The computer program product of Claim 50 wherein the computer application is
2 processable by the computer for causing the computer to:
3 embed a markup language instruction in a second message; and
4 send the second message on the real time communications protocol connection.

1 53. The computer program product of Claim 50 wherein the markup language
2 instruction is associated with a document address, and wherein the computer application is
3 processable by the computer for causing the computer to pass the document address to a
4 document acquisition apparatus.

1 54. The computer program product of Claim 50 wherein the real time
2 communications protocol connection is a real time chat communications protocol
3 connection.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 100 |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|
| 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 100 |
| 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 100 |
| 3 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 100 |
| 4 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 100 |
| 5 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 | 81 | | | | | | | | | | | | | | | | | | | |

1 56. A computer program product, comprising:
2 a computer application processable by a computer for causing the computer to:
3 form a real time communications protocol connection over a network
4 communications connection;
5 embed a markup language instruction in a message; and
6 send the message on the real time communications protocol connection; and
7 apparatus from which the computer application is accessible by the computer.

1 57. The computer program product of Claim 56 wherein the markup language
2 instruction is an html instruction.

1 58. The computer program product of Claim 57 wherein the html instruction is a
2 hyperlink instruction.

1 59. The computer program product of Claim 58 wherein the hyperlink instruction is
2 associated with a URL, and wherein the computer application is processable by the
3 computer for causing the computer to pass the URL to a Web browser.

1 60. The computer program product of Claim 57 wherein the html instruction is a
2 bold tag.

1 61. The computer program product of Claim 57 wherein the html instruction is an
2 italics tag.

1 62. The computer program product of Claim 56 wherein the real time
2 communications protocol connection is a real time chat communications protocol
3 connection.

[illegible]

1 64. A computer program product, comprising:
2 a computer application processable by a computer for causing the computer to:
3 form a real time communications protocol connection over a network
4 communications connection;
5 receive a first message on the real time communications protocol connection,
6 the first message including a markup language instruction;
7 parse the first message to identify the markup language instruction included
8 therein; and
9 display the first message in accordance with the markup language instruction
10 included therein; and
11 apparatus from which the computer application is accessible by the computer.

1 65. The computer program product of Claim 64 wherein the markup language
2 instruction is a hyperlink instruction.

1 66. The computer program product of Claim 65 wherein the hyperlink instruction is
2 associated with a URL, and wherein the computer application is processable by the
3 computer for causing the computer to pass the URL to a Web browser.

1 67. The computer program product of Claim 64 wherein the markup language
2 instruction is a bold tag.

1 68. The computer program product of Claim 64 wherein the markup language
2 instruction is an italics tag.

1 69. The computer program product of Claim 64 wherein the real time
2 communications protocol connection is a real time chat communications protocol
3 connection.

1 70. The computer program product of Claim 64 wherein the real time
2 communications protocol connection is a real time continuously open bi-directional
3 communications protocol connection.

| Variable | Mean | | SD | | t | | p | |
|----------------|---------|------|---------|------|---------|------|---------|-------|
| | Control | Case | Control | Case | Control | Case | Control | Case |
| Age | 30.5 | 30.5 | 1.2 | 1.2 | 0.0 | 0.0 | 0.999 | 0.999 |
| Gender | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.999 | 0.999 |
| Education | 12.0 | 12.0 | 1.0 | 1.0 | 0.0 | 0.0 | 0.999 | 0.999 |
| Income | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.999 | 0.999 |
| Marital status | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.999 | 0.999 |
| Occupation | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.999 | 0.999 |
| Religion | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.999 | 0.999 |
| Health status | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.999 | 0.999 |
| Family size | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.999 | 0.999 |
| Urban/rural | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.999 | 0.999 |
| Time of day | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.999 | 0.999 |
| Season | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.999 | 0.999 |
| Weather | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.999 | 0.999 |
| Time of day | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.999 | 0.999 |
| Season | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.999 | 0.999 |
| Weather | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.999 | 0.999 |
| Time of day | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.999 | 0.999 |
| Season | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.999 | 0.999 |
| Weather | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.999 | 0.999 |
| Time of day | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.999 | 0.999 |
| Season | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.999 | 0.999 |
| Weather | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.999 | 0.999 |
| Time of day | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.999 | 0.999 |
| Season | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.999 | 0.999 |
| Weather | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.999 | 0.999 |
| Time of day | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.999 | 0.999 |
| Season | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.999 | 0.999 |
| Weather | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.999 | 0.999 |
| Time of day | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.999 | 0.999 |
| Season | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.999 | 0.999 |
| Weather | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.999 | 0.999 |
| Time of day | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.999 | 0.999 |
| Season | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.999 | 0.999 |
| Weather | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.999 | 0.999 |
| Time of day | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.999 | 0.999 |
| Season | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.999 | 0.999 |
| Weather | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.999 | 0.999 |
| Time of day | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.999 | 0.999 |
| Season | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.999 | 0.999 |
| Weather | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.999 | 0.999 |
| Time of day | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.999 | 0.999 |
| Season | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.999 | 0.999 |
| Weather | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.999 | 0.999 |
| Time of day | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | | | |

ABSTRACT OF THE DISCLOSURE

-31-

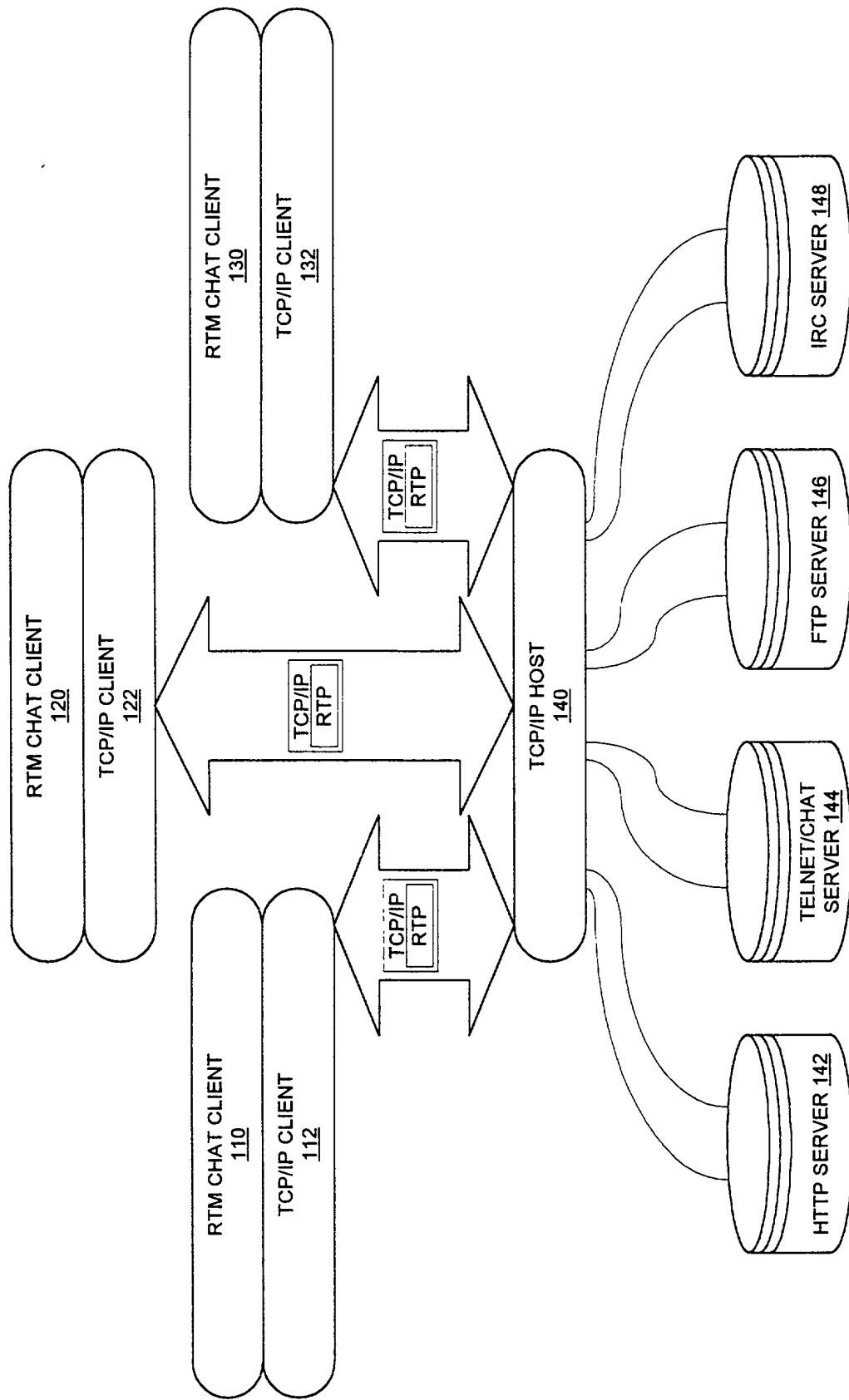


FIG. 1

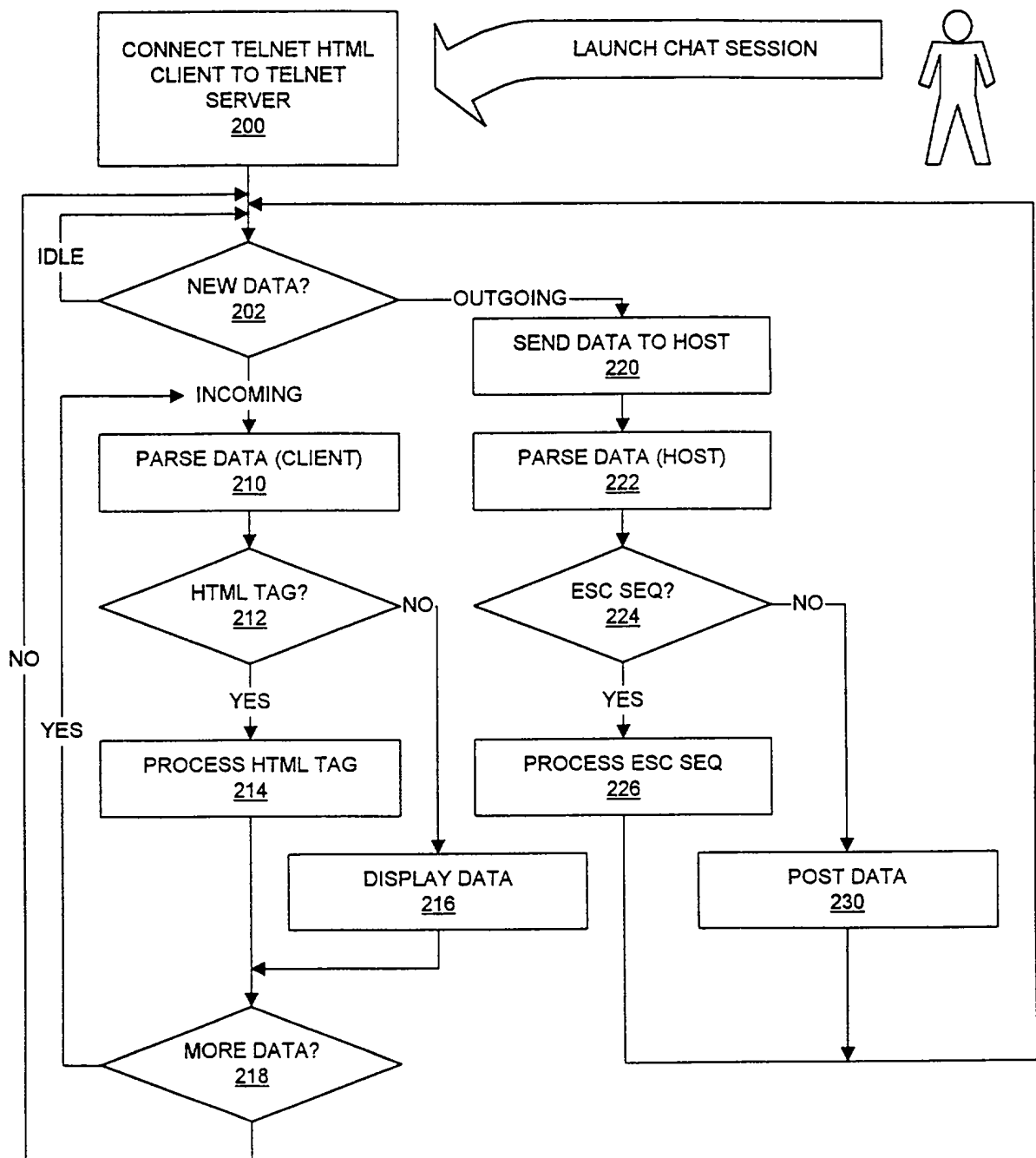


FIG. 2

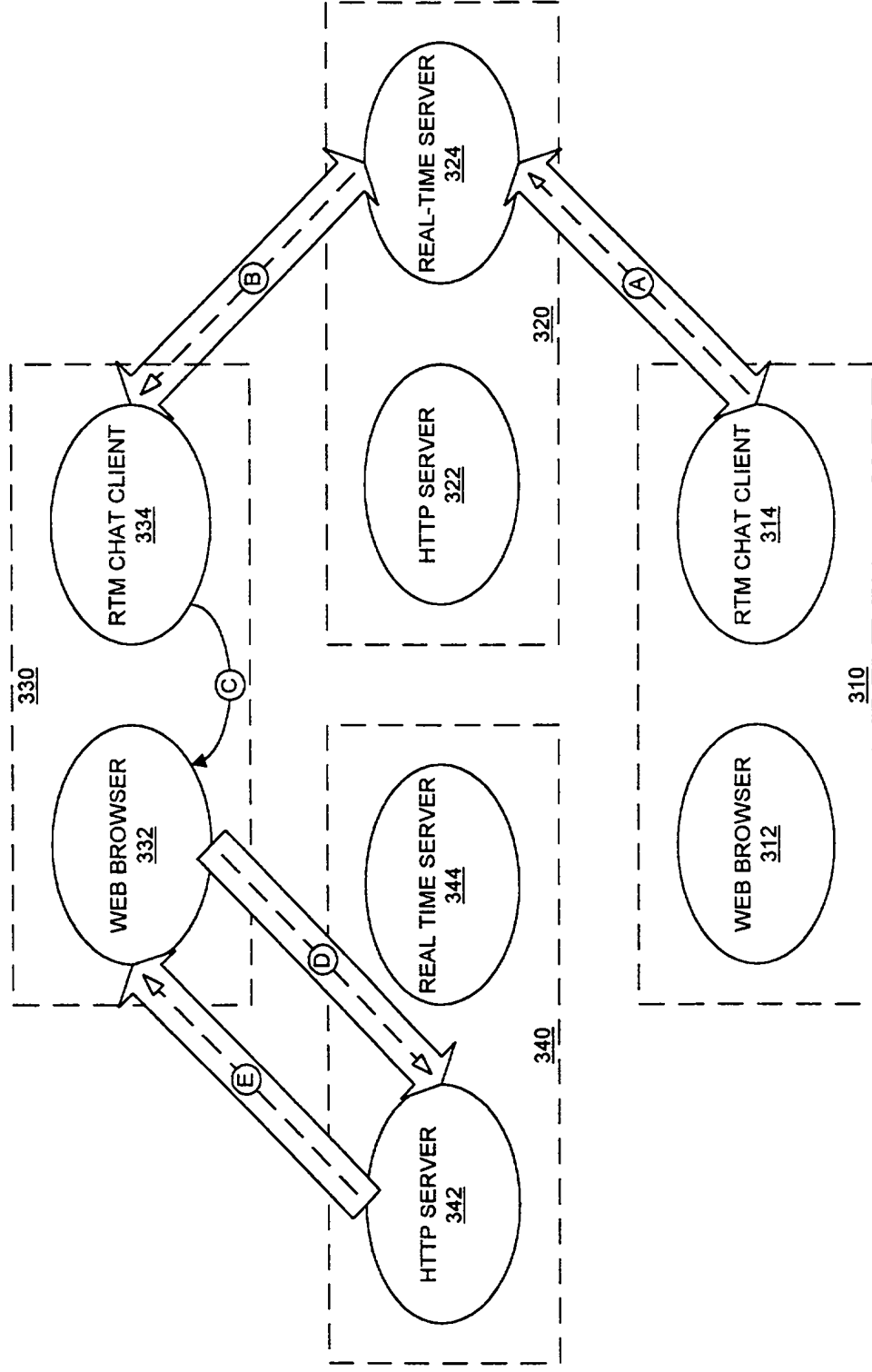
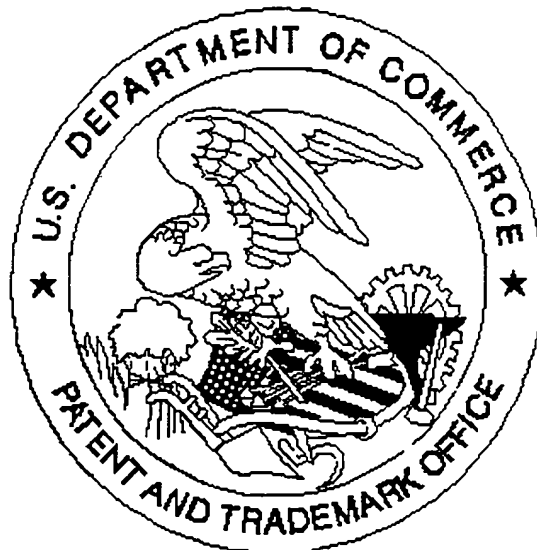


FIG. 3

United States Patent & Trademark Office

Office of Initial Patent Examination -- Scanning Division



Application deficiencies found during scanning:

1. Application papers are not suitable for scanning and are not in compliance with 37 CFR 1.52 because:
 - ☐ All sheets must be the same size and either A4 (21 cm x 29.7 cm) or 8-1/2"x 11".
Pages _____ do not meet these requirements.
 - ☐ Papers are not flexible, strong, smooth, non-shiny, durable, and white.
 - ☐ Papers are not typewritten or mechanically printed in permanent ink on one side.
 - ☐ Papers contain improper margins. Each sheet must have a left margin of at least 2.5 cm (1") and top, bottom and right margins of at least 2.0 cm (3/4").
 - ☐ Papers contain hand lettering.
2. Drawings are not in compliance and were not scanned because:
 - ☐ The drawings or copy of drawings are not suitable for electronic reproduction.
 - ☐ All drawings sheets are not the same size. Pages must be either A4 (21 cm x 29.7 cm) or 8-1/2" x 11".
 - ☐ Each sheet must include a top and left margin of at least 2.5 cm (1"), a right margin of at least 1.5 cm (9/16") and a bottom margin of at least 1.0 cm (3/8").
3. Page(s) _____ are not of sufficient clarity, contrast and quality for electronic reproduction.
4. Page(s) _____ are missing.
5. OTHER: No Declaration